# 遺伝アルゴリズムにおける交叉法に対する一考察(計算量理論)

AUTHOR(S):

柳浦, 睦憲; 茨木, 俊秀

CITATION:

柳浦, 睦憲 ...[et al]. 遺伝アルゴリズムにおける交叉法に対する一考察(計算量理論). 数理解析研究所講究録 1994, 871: 190-196

ISSUE DATE:

1994-05

URL:

http://hdl.handle.net/2433/84037

数理解析研究所講究録
第871巻 1994年 190-196


# 遺伝アルゴリズムにおける交叉法に対する一考察

京都大学工学部　柳浦睦憲　Mutsunori YAGIURA
京都大学工学部　茨木俊秀　Toshihide IBARAKI

交叉は, 遺伝アルゴリズム特有の操作であり, アルゴリズムの効果を定める重要なファクターである. 本研究では, 最適な順列を求める問題を対象として取り上げ, 提案されている色々な交叉法を比較すると共に, 交叉法の良さを測る簡単な基準を提案する. さらに, 1機械スケジューリング問題を具体例にとって計算実験を行い, 提案した基準が交叉の性能を測る有効な尺度になっていることを示す.

キーワード: 遺伝アルゴリズム, 交叉, 順序づけ問題, 1機械スケジューリング問題

## 1　序論

遺伝アルゴリズム (genetic algorithm, GA) は, 生物進化に着想を得た, 多点情報を利用する確率的アルゴリズムの1つであり, 交叉 (crossover), 突然変異 (mutation), 選択 (selection) という3つの遺伝的操作からなる [5][8][11][14]. 組合せ最適化の分野においては, アルゴリズム内部に欲張り法や局所探索法などを取り入れた方法 [13][15][22][27] や, また解集合の初期収束を避けるために, 選択を局所的な解集団ごとに行うなどの変形 [9][16][19] も研究されている.

交叉は複数個の候補解 (親と呼ぶ) から新たな解 (子と呼ぶ) を発生する操作で, GA の特徴の1つであり, アルゴリズムの性能に大きな影響をもつ [20][21]. 交叉には, 0-1 ベクトルで表現された解に対するものだけでも, 1点交叉, 2点交叉, 多点交叉, 一様交叉, 線形結合を利用した交叉など, 様々なものがある [5][14]. また, ほとんどの組合せ最適化問題においては, 解空間に制約がついており, 交叉の際にそれらの制約をみたす実行可能解を生成する必要がある. 例えば, 代表的な組合せ最適化問題の1つである行商人問題 (traveling salesman problem, TSP) においては, 解は $n$ 都市の順列でなければならないが, その条件をみたすように工夫された交叉が数多く提案されている [7][10][28](詳細は次節).

本研究では, 解空間が順列の集合であるような組合せ最適化問題に対して提案されている様々な交叉法 (以下, 順序づけ交叉と呼ぶ) を紹介し, これらを記述する一般的枠組を提供する. さらに, 良い交叉法の設計基準を明らかにするため, 遺伝アルゴリズムの簡単な枠組みを想定し, 具体的な対象として, 1機械スケジューリング問題 (single machine scheduling problem, SMP) を取り上げ, 計算実験を行った. SMP は1つの機械上で処理される仕事の最適な (即ち, コスト最小の) 順序を決定するという問題である [1][12]. その結果, 良い交叉法のみたすべき基準として, 1) 2つの親の形質の非継承要素を少なく抑える, 2) 生成可能な子の多様性を保つ, ことが重要であることを指摘し, その具体的な評価法を与えた.

## 2　従来の順序づけ交叉

本節では, これまでに提案されてきた順序づけ交叉を紹介する (欲張り法, 局所探索等を含まないものに限定). ここで紹介する交叉の多くは, TSP に対して提案されたものであるが, 一部は SMP に適するように修正して紹介する. また, 交叉の元となる親の数と生成される子の数の組合せは色々考えられるが, ここでは2つの親 $A$, $B$から1つの子 $C$が作られるものとして説明する. 説明の都合上, 順列の要素を $N=\{1,\ldots,n\}$, 順列の $i$ 番目の要素が $j$であることを$\sigma(i)=j$ (または $\sigma^{-1}(j)=i$) と表す. また, 親 $A$, 親 $B$, 子 $C$の順列を$\sigma_A, \sigma_B, \sigma_C$と記す.

**PMX** (partially mapped crossover): まず, $\{0,1\}$ をランダムに発生させて, $n$ ビットのマスク $m$ ($m(i)\in\{0,1\}$) を作る. そして, $m(i)=0$ となる各 $i$ に対し, $\sigma_C(i):=\sigma_A(i)$ とした後, $\sigma_B(j)=\sigma_A(i)$ なる $j$に対し, $\sigma_B(i)$ と $\sigma_B(j)$ の交換を行う. その後, $m(i)=1$ となる各 $i$ に対し, $\sigma_C(i):=\sigma_B(i)$ とする (図1).

上記のように, 発生したマスクに従ってどちらの親から要素を継承するかを決定する手法を一般に一様交叉 (uniform crossover) と呼ぶ. 特に高々$k$点で0と1が隣接するマスクのみを発生する場合を, $k$点交叉という. 以下では, 1点,

2 点, 一様交叉のみを考え, それぞれ PMX(1), PMX(2), PMX(U) と記す. PMX は PMX(2) として提案された [7]. 解説論文 [8][20][21] などにも紹介がある.

| | | | | | |
|---|---|---|---|---|---|
| 親A | 1 | 2 | 3 | 4 | 5 |
| 親B | 2 | 3 | 5 | 1 | 4 |
| マスク | 1 | 1 | 0 | 0 | 1 |
| 子 | 2 | 5 | 3 | 4 | 1 |

図 1: PMX(2) の 1 例.

**CX** (cycle crossover): 本方法では, 全ての $i$ に対し,

$$\sigma_C(i) = \sigma_A(i) \text{ または } \sigma_B(i) \tag{1}$$

が成り立つように$\sigma_C$を構成する. まず以下のように順列の各位置 $i$ にサイクル番号 $r(i)$ をつける (図 2). ここで, $r(i) = 0$ は番号がついていないことを表す.

1. $k := 1$, $\forall i, r(i) := 0$ とする.
2. $i_0 := \min\{i \mid r(i) = 0\}$, $i := i_0$とする.
3. $r(i) := k$, $i := \sigma_A^{-1}(\sigma_B(i))$ を $i = i_0$となるまで反復.
4. $\forall i$, $r(i) > 0$ なら終了. そうでなければ, $k := k + 1$ としてステップ 2 へ.

サイクル番号 $r(i)$ によって, 全要素はサイクル $R_k = \{i \mid r(i) = k\}$ に分割される. 同じサイクルの要素を同じ親から受け継ぐようにすれば条件 (1) を満すことができる (図 2). どちらの親から受け継ぐかを決める方法は色々考えられるが, ここでは, 各サイクル $R_k$ に対しランダムに決める方法 CX(U) (uniform より), ランダムに選んだ 1 つのサイクルのみ親 $A$ から, 残りを親 $B$から受け継ぐ方法 CX(1), 及びサイクル番号 $k$が奇数の時は親 $A$, 偶数の時は親 $B$から受け継ぐ方法 CX(A) (交互 (alternate) に取ることから) の 3 つを考察する. CX は CX(U) として提案された [20]. [8][21] などにも紹介がある.

| | | | | | |
|---|---|---|---|---|---|
| 親A | 1 | 2 | 3 | 4 | 5 |
| 親B | 3 | 4 | 5 | 2 | 1 |
| サイクル番号 | 1 | 2 | 1 | 2 | 1 |
| 子 | 1 | 4 | 3 | 2 | 5 |

図 2: CX(U) の 1 例.

**FLX** (free list crossover): まず, $n$ 要素のリスト (e.g., $(1, \ldots, n)$) を作り, 順列の各要素がこのリストの何番目であるかを左から順に定めていくことで順列を符号化する. 但し, 操作中, 既に登場した要素はリストから除いた後, 現在の要素の順位を定める. 順列$\sigma$ をこのように符号化したものを$\bar{\sigma}$ と記すと, $\bar{\sigma}(i) = \sigma(i) - |\{j \mid \sigma(j) < \sigma(i), j < i\}|$ である. $\sigma$ と$\bar{\sigma}$ は 1 対 1 に対応する. 次に, $n$ ビットの 0-1 マスク $m \in \{0,1\}^n$ を発生し, $m(i) = 0$ ならば$\bar{\sigma}_C(i) := \bar{\sigma}_A(i)$, $m(i) = 1$ ならば$\bar{\sigma}_C(i) := \bar{\sigma}_B(i)$ とすることにより, 符号化された子を作る. 出来上がった$\bar{\sigma}_C$ を複号化して子$\sigma_C$ を得る. PMX と同様, 1 点, 2 点, 一様交叉を考え, それぞれ FLX(1), FLX(2), FLX(U) と記す (図 3). FLX は FLX(1) として提案された [10].

| | σ | | | | | | $\bar{\sigma}$ | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 親A | 2 | 3 | 1 | 5 | 4 | 符号化 → | 親A | 2 | 2 | 1 | 2 | 1 |
| 親B | 3 | 1 | 5 | 4 | 2 | | 親B | 3 | 1 | 3 | 2 | 1 |
| | | | | | | 復号化 ← | マスク | 1 | 1 | 0 | 0 | 0 |
| 子 | 3 | 1 | 2 | 5 | 4 | | 子 | 3 | 1 | 1 | 2 | 1 |

図 3: リスト $(1,2,3,4,5)$ を用いた FLX(1) の 1 例.

**POPX** (partial order preserving crossover): 2 つの親に共通する 2 要素間の半順序 (partial order) を保存するような子を生成する方法である. $A$ に対し

$$D_A := \{(i,j) \mid \sigma_A^{-1}(i) \le \sigma_A^{-1}(j)\} \tag{2}$$

と定める. $D_B$, $D_C$ も同様に定義する. 2 つの親 $A$, $B$に共通する半順序 $D$は, $D := D_A \cap D_B$ と定義できる. 以下, 子 $C$として, $D$に矛盾しない (i.e., $D \subseteq D_C$) 順列の 1 つを生成する. $N$の任意の部分集合を $S \subseteq N$, $S$の $D$ に関する極小元の集合を $M_D(S) := \{i \in S \mid \forall j \in S - \{i\}, (j,i) \notin D\}$ と定義する.

1. $S := N$, $i := 1$ とする.
2. $j \in M_D(S)$ をランダムに選び, $\sigma_C(i) := j$とする.
3. $i = n$ならば終了. さもなくば, $i := i+1$, $S := S - \{j\}$ とし, ステップ 2 へ.

この方法 (POPX1 と呼ぶ) は, [22] からの発想であるが, 類似の手法が [6] にも紹介されている. また, 上記ステップ 2 において, 次の要素 $j$を $\{\sigma_A(i_A^S), \sigma_B(i_B^S)\}$ (但し, $i_A^S := \min\{i \mid \sigma_A(i) \in S\}$, $i_B^S$も同様) よりランダムに選ぶ方法も考えられる (POPX2 と呼ぶ). 図 4の例では, $N$を節点集合, $D$を有向枝集合とみなし, 半順序 $D$を表現したグラフをそえる.

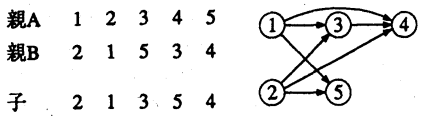


図 4: POPX1 の 1 例.

**OX** (order crossover): まず, $n$ ビットのマスク $m \in \{0,1\}^n$ をランダムに発生し, $m(i) = 0$ ならば$\sigma_C(i) := \sigma_A(i)$ とする. $S_m := \{\sigma_A(i) \mid m(i) = 0\}$ と定義すると, $D'_B := D_B - \{(i,j) \mid i \in S_m$ または $j \in S_m\}$ は $N - S_m$ の全順序である. $m(i) = 1$ の各位置に, 前から順に $D'_B$ の順序に従って要素を割り当てることにより, 子 $C$が完成する. PMX と同様, 1 点, 2 点, 一様交叉を考え, OX(1), OX(2), OX(U) と記す (図 5).

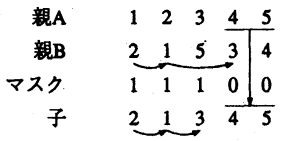


図 5: OX(1) の 1 例.

OX は [4] で OX(1) として, また独立に [15] で OX(2) として提案された. [8][20] などにも OX(2) として紹介されている. また, OX(U) は [5](p. 342~) に 2 種類の交叉法として紹介されているが, 本研究の枠組ではいずれも OX(U) に等しい.

**AEX** (alternating edge crossover): まず, 解をポインタ $p$ によって表現する. $p(i) = j$ は, 要素 $i$ の次に $j$が並ぶことを意味する. 便宜上, 出発点と終了点を併せて 0 で表すことにすると, この $p$ を用いて順列$\sigma$ は, $p(0) = \sigma(1)$, $p(\sigma(i)) = \sigma(i+1)$ $(i = 1, \ldots, n-1)$, $p(\sigma(n)) = 0$ と書ける. 親 $A$, $B$及び子 $C$のポインタ表現をそれぞれ $p_A$, $p_B$, $p_C$ と記す. 以下の操作では, まだポインタに現れていない要素の集合を $S$と表す.

1. $i := 0$, $S := N$とする.
2. $\{p_A(i), p_B(i)\} \cap S \neq \phi$ならば, この中から $j$を, さもなくば $j \in S$をランダムに選び, $p_C(i) := j$とする.
3. $i := j$, $S := S - \{j\}$ とする. $S = \phi$ならば, $p_C(i) := 0$ として終了. そうでないときはステップ 2 へ.

AEX は [10] で提案されたものであるが, 上記とはやや異なり, 次のポインタ $p_C(i)$ を $p_A(i)$, $p_B(i)$ から反復毎に交互に選び (名前, alternating の所以), 選んだものが $S$ に含まれないときは $S$ からランダムに選ぶというものである. [10][13] には, これを多少手直ししたものも提案されている (図 6).

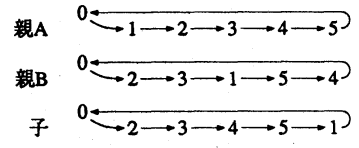


図 6: AEX の 1 例.

**ERX** (edge recombination crossover): AEX のステップ 2 において, $p_A(i), p_B(i) \in S$ のとき, $|\{p_A(p_W(i)), p_B(p_W(i))\} \cap S|$ が 0 でなく, しかも小さい方の親 $W(=A, B)$ のポインタ $p_W(i)$ を選ぶ方法である. これは, 次へのポインタが少ないものを優先することにより, ランダムなポインタ (後述の非継承要素) が選ばれる頻度を減らそうというアイディアである. ERX は [28]([5] にも紹介) で TSP に対して提案されたものであり, 本来は現在の要素 $i$ の次の要素を決める際, 2 つの親の両側で隣あっている 4 つの要素を候補とするもので, 上記とはやや異なる. [21] では, ERX の改良版も紹介されている.

**その他:** TSP に対する順序づけ交叉には, これらの他に, サブツアー交換交叉 [26], それに類似した手法 [3][15] などがある. これらについても SMP 向きに手直しして計算を行ったが, 良い結果が得られなかった (原因については [24] 参照) ので, 以下の考察には含めない. この他, [2][9][17] などにも, 上記以外の手法が挙げられているが, 上記の手法, またはそれらを組合せたものに類似しているという理由から省略する.

## 3 交叉の一般的枠組

前節の様々な交叉は全て以下のような枠組みでとらえることが出来る.

1. 親 $A$, $B$をそれぞれある構成要素の集合 $T_A$, $T_B$ として表現する. 子の構成要素集合 $T_C := \phi$とする.
2. $T_C$の新しい要素 $e$ を 1 つ選び, $T_C := T_C \cup \{e\}$ とする. $e$ は, i) $T_A \cup T_B$ の中から, または, ii) $T_C$の持つ制約条件に矛盾しない適当な要素の中から選ばれる. その後, $T_A \cup T_B$ の中で $T_C$ に矛盾するものを削除する. ここで, $e$ の選び方, 及び, i), ii) どちらの規則が用いられるかは, 交

叉法に依存する.

3. $T_C$ によって, 子が一意に表現されるようになるまでステップ 2 を繰り返す.

子の構成要素で親のどちらにも含まれないもの, 即ち $T_C - (T_A \cup T_B)$ 内の要素を, 以下, 非継承要素と呼ぶ.

上記の構成要素としては, 前節の全ての方法について

i) $T_A := \{(i, \sigma_A(i)) \mid i = 1, \ldots, n\}$

ii) $T_A := \{(i, \bar{\sigma}_A(i)) \mid i = 1, \ldots, n\}$

iii) $T_A := D_A$

iv) $T_A := \{(i, p_A(i)) \mid i = 0, \ldots, n\}$

の 4 つのどれかで表現できる ($T_B$ も同様). i) の $(i, \sigma_A(i))$ は, 親 $A$ では $i$ 番目の要素は $\sigma_A(i)$ であることを表す (他も同様). 上の 4 つを i) position-based representation (PsR), ii) free-list-based representation (FLR), iii) order-based representation (OR), iv) pointer-based representation (PtR) と呼ぶことにする.

例えば, PMX は 2 つの親を PsR で表現するとして説明できる. まず, マスク $m(i) = 0$ なる全ての $i$ について $(i, \sigma_A(i)) \in T_A$ を $T_C$ の要素とする (i.e., $\sigma_C(i) := \sigma_A(i)$ に対応). このとき, $\sigma_B(j) = \sigma_A(i)$ なる $j$ に対し, $T_B - \{(j, \sigma_B(j))\}$ とする (交換の操作の及ばないものを残す). $m(i) = 1$ なる全ての $i$ に対し, $(i, \sigma_B(i)) \in T_B$ ならばこれを $T_C$ に加える. 残りの部分は非継承要素になるが, $(i, \sigma_B(\sigma_A^{-1}(\sigma_B(i))))$ のように定める. 他の交叉も全て同様に上の枠組みで説明できる.

ところで, 上の枠組みの自然な実現法の 1 つとして, ステップ 2 の要素 $e$ の選択をランダムに行う方法が考えられる. その結果, 前節の方法とは異なる交叉法となる場合がある. 例えば PsR では, $T_A \cup T_B$ の中から $(i, \sigma_A(i))$ をランダムに選び, $T_C := T_C \cup \{(i, \sigma_A(i))\}$ としたのち, $(i, \sigma_B(i))$ 及び $\sigma_B(j) = \sigma_A(i)$ なる $j$ に対し, $(j, \sigma_B(j))$ を $T_B$ から除くという操作 ($T_C := T_C \cup \{(i, \sigma_B(i))\}$ の場合も同様) の繰り返しを行う方法となる (position-based random crossover と呼び, PsRND と記す). 同様に, order-based random crossover (ORND), pointer-based random crossover (PtRND) もランダム選択法を用いて構成できる. FLR では, FLX(U) がこれらに相当する. ポインタ $p$ が $\{0, \ldots, n\}$ の順列であることから, 解のポインタ表現に対しても CX と同様の交叉が構成できる (PtCX と呼ぶ). 但し, 部分サイクルができないように注意を払う必要がある.

親の表現法に従って交叉を分類すると, 表 1のようになる. 以上の考察から, 交叉法の違いは, 構成要素の表現法と, その選択に際して用いられるルールに帰着されるといえる.

表 1: 親の表現方法による交叉の分類.

| 親の表現法 | 交叉の方法 |
|---|---|
| PsR | PMX, CX, PsRND |
| FLR | FLX |
| OR | POPX, ORND |
| PtR | AEX, ERX, PtCX, PtRND |
| PsR+OR | OX |

## 4 GA における交叉の役割

本節では, GA において交叉が果たす役割について考察する. 2 つの親 $A$, $B$ から交叉法 $x$ (PMX(2), ERX など) によって生成され得る子の集合を $C(x; A, B) \subseteq F$ ($F$ は実行可能解全体) と表す. 即ち, 交叉法 $x$ とは, 解 $\sigma \in C(x; A, B)$ を 1 つランダムに選んで (等確率とは限らない) 出力することと解釈できる.

$C(x; A, B)$ が果たす役割の 1 つに, 見込みのありそうな解に探索を限定することが挙げられる (目標 1). 一方, むやみに探索空間を限定してしまうことには意味がなく, 見込みがあると考えられる限りはできるだけ広い範囲を探索すべきである (目標 2). この, 一見矛盾する 2 つの目標のバランスをいかにとるかが GA の成果の鍵を握っているといえる.

目標 1 を達成するために, GA の基本姿勢として, 親の構成要素をできるだけ多く受け継ぐ解を $C(x; A, B)$ に含めるという考え方がある. 具体的には, $C(x; A, B)$ に含まれる子は, 非継承要素ができるだけ少なく抑えられていることが望ましい (基準 1). GA において候補解として残っている $A$ と $B$ は, すでにかなり良い解であることが期待できるので, この基準によって, それまでの進化の結果を子に残し, 解の質を高く保つことができる. 次に, 目標 2 を達成するには, $|C(x; A, B)|$ をできるだけ大きくとることが望ましい (基準 2).

基準 1 と 2 は, 一般に, 非継承要素数を小さく抑えようとすれば $|C(x; A, B)|$ も小さくなり, $|C(x; A, B)|$ を大きくとろうとすれば非継承要素数が増えてしまう傾向にある. よって, 両者のトレードオフが重要になる. なお, 目標 1 と 2 の達成度を数値的にとらえる評価基準として, $C(x; A, B)$ 内の解のコストの平均 (基準 1′: 小さいほど良い) と標準

偏差 (基準 2′: 大きいほど良い) が挙げられる.

## 5 計算実験

**遺伝アルゴリズム** 個体数を 100 とし, 初期解は全てランダムに与える. 探索全体を通じて, 候補解中に同じ解をもつことを許さない. 選択は, 1 回の交叉毎に行い, 新しく作られた子が現在の候補解中に含まれず, しかも現在候補解中で 1 番悪い解よりもコストが改善されたとき, それと交換するという方法を取る. 突然変異は加えない. 局所探索等の解の改善手法も用いない. 本研究の目的は, 交叉法の比較を行うことにあるため, 他のオペレータとの相互干渉によって交叉法の影響が埋没してしまうことを避けるため, このような単純な枠組を定めた.

**1 機械スケジューリング問題 (SMP)** 代表的なスケジューリング問題の 1 つであり, これまで多くの研究がある [1][12]. 与えられた $n$ 個の仕事 $N = \{1, \ldots, n\}$ を 1 台の機械で処理するが, 機械は 1 度に 1 つの仕事しか処理できず, 処理の中断, 処理間の無駄時間は許されない. 従って, スケジュールは $n$ 個の仕事の順列 $\sigma$ を与えることで決定される. 各仕事 $i$ は処理時間 $p_i$ を要し, 完了時刻が $c_i$ であるときに $g_i(c_i)$ のコストがかかる. $c_i = \sum_{j=1}^{\sigma^{-1}(i)} p_{\sigma(j)}$ である. このとき,

$$\mathrm{cost}(\sigma) = \sum_{i \in N} g_i(c_i) \rightarrow \min \tag{3}$$

を実現する $\sigma$ を求める問題である. コストの例には,

$$g_i(c_i) = h_i \max\{d_i - c_i, 0\} + w_i \max\{c_i - d_i, 0\}$$

を用いた. ここに $d_i$ は仕事 $i$ の納期, $h_i, w_i$ はそれぞれ仕事 $i$ の進み, 遅れに対するペナルティの重みである. この定義に限らず, 多くの $g_i(c_i)$ について SMP の NP 困難性が知られている.

**実験結果** 計算実験は, SUN SPARC station IPX 上で行い, C 言語を用いた. 問題例は, [12] で用いられた方法に従って, $n = 35, 100$ に対し 10 問ずつランダムに発生した. これは, SSDP 法 [12] によって厳密解の得られる問題例 (最大 $n = 35$) と, 比較的サイズの大きい問題例 ($n = 100$) を調べるためである.

表 2: 様々な交叉法の比較.

| 表現法 | 交叉法 $x$ | 最良解からの誤差 (%) | | 非継承要素の割合 (%) | | $\lvert C(x)\rvert$ | $C(x;A,B)$ 内の解の平均 | | $C(x;A,B)$ 内の解の標準偏差 | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | $n=35$ | $n=100$ | $n=35$ | $n=100$ | | $n=35$ | $n=100$ | $n=35$ | $n=100$ |
| PsR | CX(A) | 46.5 | 97.7 | 0.0 | 0.0 | $O(1)$ | 0.062 | −0.001 | 1.5 | 1.4 |
| | CX(1) | 47.9 | 97.0 | 0.0 | 0.0 | $O(\log n)$ | 0.007 | 0.018 | 1.2 | 1.1 |
| | CX(U) | 61.9 | 100.0 | 0.0 | 0.0 | $O(n)$ | 0.028 | −0.004 | 1.2 | 1.2 |
| | PMX(1) | 54.4 | 136.3 | 14.9 | 15.9 | $O(n)$ | 0.171 | 0.187 | 1.1 | 1.2 |
| | PMX(2) | 16.3 | 47.2 | 16.1 | 16.5 | $O(n^2)$ | 0.713 | 0.798 | 1.5 | 1.4 |
| | PMX(U) | 8.0 | 27.7 | 22.8 | 24.2 | $2^{O(n)}$ | 0.806 | 0.807 | 1.7 | 1.4 |
| | PsRND | 9.4 | 25.9 | 12.7 | 13.2 | $2^{O(n)}$ | 0.467 | 0.426 | 1.6 | 1.4 |
| FLR | FLX(1) | 110.2 | 181.2 | 0.0 | 0.0 | $O(n)$ | 0.733 | 0.649 | 1.4 | 1.4 |
| | FLX(2) | 82.3 | 152.6 | 0.0 | 0.0 | $O(n^2)$ | 1.048 | 0.882 | 1.6 | 1.4 |
| | FLX(U) | 56.5 | 100.8 | 0.0 | 0.0 | $2^{O(n)}$ | 2.069 | 1.880 | 1.9 | 1.6 |
| OR | POPX2 | 105.9 | 205.8 | 0.0 | 0.0 | $2^{O(n)}$ | 0.023 | −0.020 | 0.5 | 0.3 |
| | POPX1 | 67.6 | 160.9 | 0.0 | 0.0 | $2^{O(n)}$ | −0.155 | −0.356 | 0.8 | 0.5 |
| | ORND | 8.3 | 12.3 | 6.5 | 7.8 | $2^{O(n)}$ | 0.549 | 0.561 | 1.6 | 1.3 |
| PtR | PtCX | 116.4 | 206.9 | 0.0 | 0.0 | $O(n)$ | 0.608 | 0.442 | 1.4 | 1.3 |
| | ERX | 76.8 | 122.6 | 17.2 | 17.2 | $2^{O(n)}$ | 2.842 | 2.937 | 2.0 | 1.8 |
| | AEX | 79.4 | 135.2 | 19.5 | 19.5 | $2^{O(n)}$ | 2.905 | 3.089 | 2.1 | 1.8 |
| | PtRND | 16.2 | 56.9 | 14.0 | 13.7 | $2^{O(n)}$ | 2.429 | 3.120 | 2.1 | 1.7 |
| PsR + OR | OX(1) | 110.1 | 190.9 | 0.0 | 0.0 | $O(n)$ | −0.077 | −0.117 | 0.9 | 1.0 |
| | OX(2) | 10.7 | 36.0 | 0.0 | 0.0 | $O(n^2)$ | 0.379 | 0.474 | 1.3 | 1.2 |
| | OX(U) | 1.4 | 3.8 | 0.0 | 0.0 | $2^{O(n)}$ | 0.135 | 0.032 | 1.5 | 1.2 |
| | RND | 182.0 | 224.5 | 100.0 | 100.0 | $O(n!)$ | 3.440 | 3.170 | 2.1 | 1.7 |

表 2に以下の値を示す. i) 交叉回数を $n = 35$ のときは 10000 回 (PtR のみ 30000 回), $n = 100$ のとき 30000 回 (PtR のみ 90000 回) で打ち切ったときの解の質 (但し, 解の

質は，これまで実験中 [22] に求まった最良の解からの誤差の平均 (%) で測っている ($n = 35$ では最適解からの誤差)). ii) 子に含まれる非継承要素の割合 (%). iii) $A$, $B$をランダムに発生したときの $|C(x;A,B)|$ の期待値 $|C(x)|$(オーダーで示す. 詳細は [23]). iv) 親 $A$, $B$を固定したとき，交叉によって生成される解$\sigma \in C(x;A,B)$ の親のコストの平均値 $m_{AB} = (\mathrm{cost}(\sigma_A)+\mathrm{cost}(\sigma_B))/2$ からのずれを正規化した値 $(\mathrm{cost}(\sigma) - m_{AB})/(\frac{1}{2}|\mathrm{cost}(\sigma_A) - \mathrm{cost}(\sigma_B)|)$ の平均値. v) 同様に親 $A$, $B$に対して正規化したコスト値 $\mathrm{cost}(\sigma)$ の標準偏差. 探索の打ち切り回数は，探索の進行に伴う解の改善の様子を調べ (詳細は [23])，これ以上大きな改善が見込めなくなった時点とした. 交叉法は，表現法毎に (表1参照)，上から順に $|C(x)|$ の小さい順に並べてある. RND は，独立にランダムな解を発生する操作を反復するランダムサーチであり，GA の効果を示す基準を提供する.

データ ii) は，独立なサンプル 10000 個に対する結果である. データ iv), v) において，親 $A$, $B$は初期解中の最良解および 10 番目に良い解とし，各問題に対し 1000 個サンプルを取り，10 問ずつの平均を取った.

表より，非継承要素の割合は，RND を除き全ての場合について比較的小さな定数となることが確かめられる. このことは理論的にも確認できている. また，解の表現法毎にみると，$|C(x)|$ のオーダーが大きいほど解の質が高くなる傾向が明確に表れている. なお，$|C(x)| = 2^{O(n)}$と表記したほとんどの交叉法において，$|C(x)| \approx O(2^n)$ であり，あまり大きな差はないが，OR の 3 つでは，POPX2 が $O(2^n)$, POPX1 が $O(2^{2n})$, ORND が $O(2^{4n})$ のようにかなり差があることが観測されている.

$C(x;A,B)$ 内の解の平均は，解の表現方法が等しければほぼ同程度の値となり，FLR, PtR の交叉法ではかなり大きいが，他は比較的小さな値になっている. また，表現法毎にみると，非継承要素の割合が大きくなるに従ってやや悪くなる傾向にある. 標準偏差は，POPX の 2 つでは他に比べて極端に小さく，ERX, AEX, PtRND ではやや大きいが，それ以外のものでは大きな差はないことが分かる.

以上の結果は，前節の考察を裏付けるものとなっている. 即ち，非継承要素数は RND を除いていずれの交叉法でも十分小さい (基準 1) ので，同じ表現法中では，$|C(x)|$ が大きいほど解の質が高い (基準 2). 具体的には，$|C(x)| \geq 2^{O(n)}$ 程度が必要と思われる. RND は，$|C(x)|$ は大きいが非継承要素数が極端に大きいので，良い結果を得ない (基準 1). $C(x;A,B)$ 内の解の平均が悪い交叉法 (FLR, PtR のもの) では，あまり良い結果が得られない (基準 1′). POPX の 2 つは，$C(x;A,B)$ 内の解の標準偏差が他に比べて極端に小さいため，良い結果を得ない (基準 2′).

全体としては，OX(U) 及び，子の構成要素の選び方のルールをランダム選択法としたもの (e.g., PsRND, ORND) が良い結果を得ている. これらは，比較的自然に構成でき，しかも提案した基準を満たす交叉法となっている. 逆に，1 点交叉や 2 点交叉などのように，問題の本質に関係のないルールを用いて $|C(x)|$ を小さくすることには意味がないことも結論できる.

以上の結果より，良い交叉法を構成するためには，提案した基準が満されていることが望ましいが，一般に $|C(x)|$ 及び $C(x;A,B)$ 内の解の標準偏差を大きくすると，非継承要素数及び $C(x;A,B)$ 内の解の平均も大きくなる傾向にあるため，これらの基準のトレードオフが大切である. このための指針として，1) 非継承要素の割合を小さく抑えながら (交叉法を第 3章の枠組みのようにとらえ，子の構成プロセスにおいて可能なかぎり親の構成要素を用いることで，おおむね達成できると考えられる)，$|C(x)|$ を $2^{O(n)}$ となる程度まででできるだけ大きくとる，2) $C(x;A,B)$ 内の解の平均が良くなるように，問題の性質をうまくとらえた解の表現方法を用いる，の 2 つが挙げられる. また，OX のように，2 つの表現方法を組合わせた方法も有効である場合がある. 指針 1) は交叉を設計する際にある程度予測できるのに対し，指針 2) は構成した交叉法を実行してみないと分からないデータである. よって，以上の指針は，表現法を固定した場合には有効であると思われるが，どの表現法が望ましいかについては解くべき問題の構造を十分理解することが重要であると言わざるを得ない.

## 6 まとめ

GA の交叉法に着目し，解空間が順列であるような問題に対するオペレータに限って，今までに提案されてきた諸法を一般的枠組みの下で比較検討した. その結果，交叉法の良さは，親 $A$, $B$から交叉法 $x$ によって生成され得る子の集合 $C(x;A,B)$ が持つ性質に基づいて簡単な基準で評価できることが分かった. これらは，GA における交叉法の設計において有効な指針となることが期待される.

本研究では，突然変異を用いず，淘汰圧の高い淘汰法を用いたため，候補解に多様性を持たせる役割が交叉に求められる結果となり，$|C(x)|$ の大きい交叉法が望ましいとい

う傾向が観測された. GA において, 候補解に多様性を持たせる方法には, この他に, 突然変異を加える, 個体数を大きくとる, 淘汰圧の小さい淘汰法を用いる, など色々挙げられる. これらの方法との比較については, 今後の課題としたい. また, 組合せ最適化問題に対する高性能な GA を構成するためには, 局所探索法などの解の改善手法を組合せることが不可欠であると思われ [22][27], この点に関する研究も今後必要である.

GA は汎用性が高く, アルゴリズム内部に様々な要素を含むため, それらに色々な工夫を加えることで性能を改善できる可能性がある. このロバスト性は GA の一つの魅力であるが, 一方, GA を利用する立場からは, アルゴリズムは出来るだけ簡明であることが望ましいという要望がある. この意味で, GA の枠組を出来るだけ単純化し, どの部分がアルゴリズムの性能にどの様な影響を与えるかを見え易くする研究も重要である. 本研究は, この様な目的に貢献することを目指したものである.

**謝辞** 適切な助言をいただいた永持仁助教授はじめ日頃熱心に指導いただく研究室の諸氏に厚く御礼申し上げます. なお, 本研究は一部文部省科学研究費によるものである.

## 参考文献

[1] K. R. Baker and G. D. Scudder, "Sequencing with Earliness and Tardiness Penalties: A Review," *Operations Research* **38**, No. 1, 22–36, 1990.

[2] J. N. Bhuyan, V. V. Raghavan and V. K. Elayavalli, "Genetic Algorithm for Clustering with an Ordered Representation," *Proc. 4th ICGA*, 408–415, 1991.

[3] R. M. Brady, "Optimization Strategies Gleaned from Biological Evolution," *Nature* **317**, 804–806, 1985.

[4] L. Davis, "Applying Adaptive Algorithms to Epistatic Domains," *Proc. 9th Int. Joint Conf. on Artificial Intelligence (IJCAI)*, 162–164, 1985.

[5] L. Davis, *Handbook of Genetic Algorithms*, Van Nostrand Reinhold, 1991.

[6] B. R. Fox and M. B. McMahon, "Genetic Operators for Sequencing Problems," *Foundations of Genetic Algorithms (FOGA)*, 284–300, 1991.

[7] D. E. Goldberg and R. Lingle, "Alleles, Loci, and the Traveling Salesman Problem," *Proc. 1st ICGA*, 154–159, 1985.

[8] D. E. Goldberg, *Genetic Algorithms in Search, Optimization and Machine Learning*, Addison-Wesley, 1989.

[9] M. Gorges-Schleuter, "ASPARAGOS An Asynchronous Parallel Genetic Optimization Strategy," *Proc. 3rd ICGA*, 422–427, 1989.

[10] J. Grefenstette, R. Gopal, B. Rosmaita and D. Van Gucht, "Genetic Algorithms for the Traveling Salesman Problem," *Proc. 1st ICGA*, 160–168, 1985.

[11] J. Holland, *Adaptation in Natural and Artificial Systems*, The University of Michigan Press, 1975., and MIT Press, 1992.

[12] T. Ibaraki and Y. Nakamura, "A Dynamic Programming Method for Single Machine Scheduling," *European J. of Operational Research* (to appear).

[13] P. Jog, J. Y. Suh and D. Van Gucht, "The Effects of Population Size, Heuristic Crossover and Local Improvement on a Genetic Algorithm for the Traveling Salesman Problem," *Proc. 3rd ICGA*, 110–115, 1989.

[14] 北野, 遺伝的アルゴリズム, 産業図書, 1993.

[15] H. Mühlenbein, M. Gorges-Schleuter and O. Krämer, "Evolution Algorithms in Combinatorial Optimization," *Parallel Computing* **7**, 65–85, 1988.

[16] H. Mühlenbein, "Parallel Genetic Algorithms, Population Genetics and Combinatorial Optimization," *Proc. 3rd ICGA*, 416–421, 1989.

[17] H. Mühlenbein, "Parallel Genetic Algorithms in Combinatorial Optimization," *Computer Science and Operations Research*, Pergamon Press, 441–453, 1992.

[18] 西川, 玉置, "ジョブショップ型スケジューリング問題に対する遺伝アルゴリズムの一構成法," 計測自動制御学会論文集, 27 巻, 5 号, 593–599, 1991.

[19] 西川, 玉置, "近傍モデルによる遺伝アルゴリズムの並列化手法とそのジョブショップ・スケジューリング問題への適用," 計測自動制御学会論文集, 29 巻, 5 号, 589–595, 1993.

[20] I. M. Oliver, D. J. Smith and J. R. C. Holland, "A Study of Permutation Crossover Operators on the Traveling Salesman Problem," *Proc. 2nd ICGA*, 224–230, 1987.

[21] T. Starkweather, S. McDaniel, K. Mathias, D. Whitley and C. Whitley, "A Comparison of Genetic Sequencing Operators," *Proc. 4th ICGA*, 69–76, 1991.

[22] M. Yagiura, "Genetic Algorithms for Solving Some Combinatorial Optimization Problems," master's thesis, Department of Applied Mathematics and Physics, Faculty of Engineering, Kyoto University, 1993.

[23] 柳浦, 茨木, "順序問題における遺伝的交叉法に対する一考察," (電気学会論文誌に投稿中).

[24] 柳浦, 永持, 茨木, "サブツアー交換交叉に対する 2 つのコメント," (人工知能学会誌に投稿予定).

[25] T. Yamada and R. Nakano, "A Genetic Algorithm Applicable to Large-Scale Job-Shop Problems," *Proc. 2nd Int. Workshop on Parallel Problem Solving from Nature*, 281–290, 1992.

[26] 山村, 小野, 小林, "形質の遺伝を重視した遺伝的アルゴリズムに基づく巡回セールスマン問題の解法," 人工知能学会誌 **7**, No. 6, 117–127, 1992.

[27] N. L. J. Ulder, E. Pesch, P. J. M. Van Laarhouven, H-J. Bandelt and E. H. L. Aarts, "Genetic Local Search Algorithms for the Traveling Salesman Problem," *Proc. 1st Int. Workshop on Parallel Problem Solving from Nature* 1990.

[28] D. Whitley, T. Starkweather and D. Fuquay, "Scheduling Problems and Traveling Salesmen: The Genetic Edge Recombination Operator," *Proc. 3rd ICGA*, 133–140, 1989.